(U) Ask Raul: Damaged Data

FROM: Raul
A DNI Analyst
Run Date: 01/18/2005

Dear Raul,
(C) I get an awful lot of damaged attachments in my DNI. Is there anything I can do, within reason, to help with this?

Betty

Betty,

(U) The short answer is, "Yep." It would be nice though, if something could be done about the huge amount of damaged data we haul in. Some of the material we get is damaged for valid reasons but a lot is damaged due to things *we* are doing (or not doing). So, I'll give you a hint or two on how to prevent problems and a few on how to solve some of them on your own.

(S) Several years ago over in the Office of Russia we had been waiting for what seemed like forever to get collection from a particular link coming from a satellite parked over Vladivostok. No sooner had we started getting the collection than we were let down. On the positive side the sessions were, for the time, huge and looked very juicy but every single one of them was corrupted.

(C) Most folks just blew it off as bad luck and went on about their business but when I looked at the SRI, which didn't include the signal strength, I knew exactly what was wrong. You see, the decision had been made to task the signal at Bad Aibling, a mere 1/3rd the way around the world from Vlad and of course the antenna was depressed to such a low elevation that the signal was being picked up through the surrounding pine forests and in the process was dropping out. In other words, we were suffering from a self-inflicted SIGINT wound. Misawa would have been a much better location to collect this signal from but...

(C) If BA had been using a coat hanger for an antenna things wouldn't have been much different than what was effectively pointing the big one at the ground! So, rule number one is to make sure you are collecting your signal from the right site.

(S) Another problem is our equipment. Things don't always work as advertised and sometimes things just get fouled up. Not long ago an analyst asked Raul to look at some traffic that was "damaged". But, it wasn't damaged at all. It was unprocessed TCP. As it turned out someone had accidentally misconfigured the protocol processor at the site and had it stop once it had finished with IP. Oops! Interestingly, this problem had been going on for about six weeks and no one had noticed.

(U//FOUO) In another case, a similar misconfiguration had been in place for well over five years and no one caught it. So, rule number two is to make sure things are working. Relying on the traffic

SERIES:
(U) Ask Raul - Answers to DNI Questions

fairies will get you in trouble.

(U) Remember, damage doesn't just happen, something causes it and it is definitely much better to prevent it than to try and fix it.

(U) That said, here are some things you can do if you find yourself sitting there pulling your hair out and crying over your damaged base64 encoded Word document from Osama.

(C) First, do another search. For the love of Pete, please do another search! I can't count the times folks have contacted me about a damaged file they have and all I have done is do another query for the file. On the one hand, it makes Raul look like a genius but on the other, it sure makes me wonder about our analysts.

(U//FOUO) Usually, you've got all the information you need: file name, sender, recipient, etc. You'd be amazed, or maybe not, at how much duplicate traffic we have in the system. If you are lucky, you'll find a second copy of your attachment in perfect condition. If not perfect, you can often patch a good attachment together from the pieces of several bad ones. This is a bit time consuming but very rewarding when you create your own little Frankenstein's monster which displays perfectly. A simple text editor is the ideal tool for this type or work.

(C) Second, if you can't find another copy of your wounded attachment, give it an examination. Are there lots of errors or just a few? Do the errors look like whoppers or is there just a byte or two missing? What kind of file is it -- Word, Excel, PDF, ZIP, etc.? It is much easier to recover text than binary material.

(U//FOUO) And lastly, do you have any way of handling the damaged file once you do whatever voodoo you do to it? If you've got so many errors it looks like someone was shooting the traffic with a machine gun, then it is probably best to just say a little prayer and bury it. If not, prep for surgery!

(S) Third, get hopping. The best way to demonstrate this is to just give an example. Here we go! Let's say I get some "collection" and try to open the attachment, a Word document, but it all goes kaboom! I've tried some additional searches and turned up nothing but goose eggs. So, I look at the raw file and see something like this inside it amongst all the base64 encoded lines:

I immediately see that I'm missing a few bytes from the third line. If I go ahead and decode this, I'll end up with the following mess:

Not good. Now, I know the base64 alphabet covers the characters A-Z, a-z, 0-9, +, / and = and since = is just a pad, we have to start guessing. Since I'm expecting text to be here I'm going to insert two /'s at the end of the bad line to get the line length back in place and then base64 decode it. Here's how it looks now:

and when we decode it:

(U//FOUO) Quite obviously this made a whole lot of difference. Had this been some kind of binary file (i.e. image) I'd have used A's instead of the /'s. I use the A & / as my preference but you can use other characters if you wish. Also, if you look at your data and see something like this:

BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Then obviously the correct thing to do would be to pad out the short line with three more B's. Use your noodle and you'll usually be safe.

(S) Something else to remember here is the number 4. Whatever you have in base64 should be a multiple of 4. Do not count any carriage returns or linefeeds. If you fix something but don't make sure the line length is a multiple of 4, things are not going to work.

(U) Likewise, it is possible the error could be in the center of the line somewhere and not conveniently located at the very end as in the example above. Let's take the same example from above but this time we'll put the damage somewhere in the middle so that it looks like this:

(U)This time when I add the two /'s to the end of the line I just get the mess below which tells me the error is somewhere within the short line:

(U) So, I start decoding the third line in chunks of 4 bytes. I'll pick the first half or so of the line and decode it. Remember, I need to pick a multiple of 4. If things come out correctly, then I'll know the error is somewhere in the second half. If not, it has to be in the first half. Simple enough. We take the first 36 bytes and it looks like this.

and when decoded:

(U) This is obviously messed up and since it is right at the end of the 36 bytes, I'll concentrate on the inserting my patching characters somewhere between bytes 33 & 36. Lucky us though, since there is something readable underneath, I can insert the two missing characters in any of the positions and it will decode close enough for me to guess even if it isn't perfect. Watch:

(U) If you can't figure out what the garbled word is supposed to be here, then it's time to get out of the analysis biz. The key thing to remember is that everything, while base64 encoded, has to be a multiple of 4.

(C) So, there you go. I've just barely scratched the surface here on repairing data but it can be done and it doesn't take a degree in rocket science or fancy tools. All the tools you need are available via the corporate toolset on everyone's desktop (at least if you are

in S2). If you'd like to let an automated process help you out, you can always try running your damaged files through Rocksolid ([redacted]). You won't find the flexibility you'd have in doing it by hand but it is a start.

Enjoy!
